Proceedings of the 2024 Joint Rail Conference
JRC2024
May 13-15, 2024, Columbia, South Carolina

JRC2024-130045

FEATURE EXTRACTION FROM VIBRATION SIGNATURE REQUIRED FROM RAILROAD BEARING ONBOARD CONDITION MONITORING SENSOR MODULES

Kevin Quaye
University of Texas Rio Grande Valley
Edinburg, TX

Ping Xu
University of Texas Rio Grande Valley
Edinburg, TX

Dimah Dera
Rochester Institute of Technology
Rochester, NY

Heinrich Foltz
University of Texas Rio Grande Valley
Edinburg, TX

Constantine Tarawneh
University of Texas Rio Grande Valley
Edinburg, TX

Alberto Diaz
University of Texas Rio Grande Valley
Edinburg, TX

ABSTRACT

From 2013 to 2022, 1671 derailments have been reported by the Federal Railroad Administration (FRA), 8.2% of which were due to journal bearing defects. The University Transportation Center for Railway Safety (UTCRS) designed an onboard monitoring system that tracks vibration waveforms over time to assess bearing health through three analysis levels. However, the speed of the bearing, a fundamental parameter for these analyses, is often acquired from Global Positioning System (GPS) data, which is typically not available at the sensor location. To solve this issue, this paper proposes to employ Machine Learning (ML) algorithms to extract the speed and other essential features from existing vibration data, eliminating the need for additional speed sensors. Specifically, the proposed method tries to extract the speed information from the signatures that are embedded in the Power Spectral Density (PSD) plot, which enables rapid real-time analysis of bearings while the train is in motion. The rapid extraction of data could be sent to a cloud accessible by train dispatchers and railcar owners for assessment of bearings and scheduling of replacements before defects reach a dangerous size. Eventually, the developed algorithm will reduce derailments and unplanned field replacements and afford rail stakeholders more cost-effective preventive maintenance.

Keywords: machine learning, power spectral density, feature extraction

1. INTRODUCTION

In recent years the railway industry has experienced a number of railcar derailments predominantly due to failure of bearing and wheel assemblies. Hence, safety, reliability, efficiency, and performance of bearings have become the main concerns in the industry. Condition monitoring and fault diagnosis of railcar bearings have become very important but are often complex, time-consuming, and labor-intensive [1]. Hot-Box Detectors (HBDs) and the Trackside Acoustic Detection System (TADS^{TM}) are the current bearing condition monitoring systems. HBDs use non-contact infrared sensors to detect abnormal temperatures of the bearings as they pass over the detector. Bearings with temperatures around 94°C above ambient conditions are removed for inspection. However, from 2010 to 2018, 124 defective bearings were not detected by HBDs [2]. TADS^{TM} uses microphones to detect high-risk bearings by listening to its acoustic sound vibrations, but they are not very reliable since there are less than 30 active systems in the U.S and derailments may occur before encountering any of these systems [2]. The University Transportation Center for Railway Safety (UTCRS) over the past years have carried out research to solve the issues of bearing failure by determining the best indicators of bearing health.

An advanced algorithm has been developed by researchers at the UTCRS that monitors bearings via temperature and vibration measurements. The vibration measurements are recorded by accelerometers on the bearing adapters to determine if the bearing is defective, the location of the defect and the approximate size of the defect [2]. However, research has shown that vibration signals collected and processed by on-board sensors are often contaminated by noise and sometimes unusable for vibration analysis of the bearing. Properties such as the frequency, amplitude and harmonics of these signals can go undetected without the help of special techniques [1].

Feature extraction is a process in machine learning (ML) and signal processing where relevant and important information is selected, identified, or transformed from raw data. In the context of detecting machine faults from vibration signals, feature extraction techniques play a pivotal role in enhancing the signal-to-noise ratio. While there are experimental or physics-driven methods to carry out vibrational analysis, inherent errors in these calculations can arise, stemming from sensor faults, material wear, fouling, slipping, defects, or excessive noise in vibration data. To overcome these challenges, data-driven methods leveraging real-time sensor data have emerged as effective tools. These methods involve constructing statistical models using abundant vibration data captured every few seconds through data acquisition systems such as the onboard condition monitoring sensor module. At UTCRS, researchers have recognized the potential of harnessing this wealth of data, particularly in extracting features such as speed from vibration signals. This exploration has prompted research focusing on vibration feature extraction techniques within the frequency domain using ML techniques.

Motivated by the universality of kernel methods in function approximation, in this study, we propose to use the kernel-based methods, specifically, Kernel Ridge Regression (KRR), to extract the speed feature from the vibration signals [3]. The kernel function utilized in this framework is the polynomial kernel that is frequently used in many kernelized algorithms such as Support Vector Machines (SVMs) [4]. We formulate the feature extraction process as an optimization problem in which we use KRR to approximate a nonlinear function that describes the relationship between the speed and the vibration data. Notably, compared with neural networks [5], another popular method in ML areas, kernel methods work well for smaller datasets with a moderate number of features and offer more interpretability.

This paper seeks to offer a new approach to finding feature extraction methods that make it possible to extract the speed of railroad bearings from the vibration signals. This would be a departure from the conventional methods of carrying out vibrational analysis on railroad bearings.

2. PRELIMINARY WORK

The UTCRS has designed an onboard monitoring system that uses sensors to track vibration waveforms over time, to obtain a direct and more accurate indicator of bearing health [2]. The resulting advanced Defect Detection Algorithm (DDA) is shown in FIGURE 1. The data collected by the onboard sensors is used for Level 1, 2 and 3 analyses of the bearings. Level 1 analysis determines whether the bearings are defective or healthy based on the root mean square (RMS) of the vibration data being greater or less than the average threshold for defect-free bearings. Level 2 analyzes the power spectral density (PSD) plot that is generated, to identify the defect type. Level 3 determines the defect size using a library of previously measured defects.

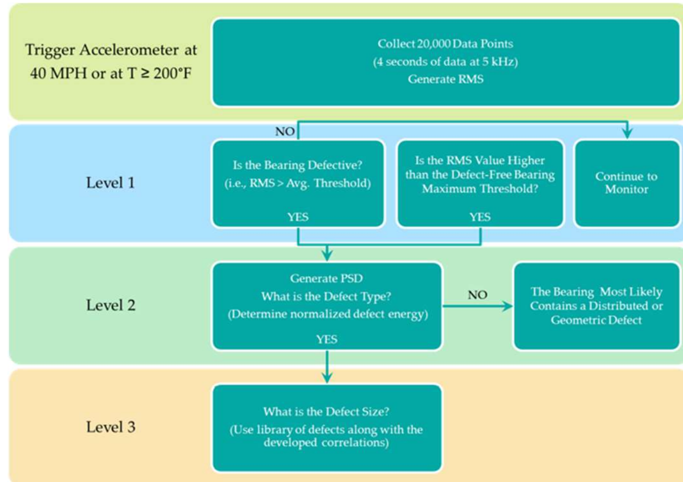

FIGURE 1: DEFECT DETECTION ALGORITHM [2]

Specifically, DDA uses frequency-domain analysis and relies on the rotational speed of the bearings ω_o through the vibration signals generated by the onboard sensor module while the train is in motion. Using Eq. (1), the PSD plots can be created, in which a faulty component will exhibit a power spike at the corresponding defect frequency, while a healthy bearing will exhibit no significant power changes. In order to correctly classify the sort of defect found within the bearing, certain frequencies are essential, which can be tracked based on the rotational speed of the bearings, as shown in Eq. (2) – Eq. (7), among which, Eq. (5) - (7) represent three different tapered roller bearings defect frequencies related to faulty outer rings, inner rings, and rollers, respectively. Examples of each kind of bearing condition, along with the matching defect frequency and its harmonics, are displayed in Figure 2 [2]. The notations used in DDA are given in TABLE 1.

TABLE 1: NOTATIONS FOR DDA.

SYMBOL	DEFINITION
$X(f)$	Frequency Function (Hz)
ω_o	Rotational speed (rad/s)
ω_{cone}	Rotational frequency of cone (Hz)
ω_{cage}	Rotational frequency of cage (Hz)
ω_{in}	Defect frequency of cone (Hz)
ω_{out}	Defect frequency of cup (Hz)
$\omega_{rolldef}$	Defect frequency of roller (Hz)
R_{roller}	Roller Radius (m)
R_{cone}	Cone Radius (m)
R_{cup}	Cup Radius (m)
D_{roller}	Roller Diameter (m)

$$PSD = \int_{-\infty}^{\infty} |X(f)|^2 df \tag{1}$$

$$\omega_{cone} = \omega_o \quad (2)$$

$$\omega_{cage} = \left(\frac{R_{cone}}{R_{cone} + R_{cup}}\right)\omega_{cone} \quad (3)$$

$$\omega_{roller} = \left(\frac{R_{cone}}{D_{roller}}\right)\omega_{cone} \quad (4)$$

$$\omega_{out} = 23\omega_{cage} \quad (5)$$

$$\omega_{in} = 23(\omega_{cone} - \omega_{cage}) \quad (6)$$

$$\omega_{rolldef} = \left(\frac{R_{cup}}{R_{roller}}\right)\omega_{cage} \quad (7)$$

Noticeable, Eq. (5) – Eq (7) show that the defect frequencies are all functions of the rotational speed ω_o. In current practice, the rotational speed is often acquired from Global Positioning System (GPS) data, which is typically not available at the sensor location. Hence, if we can extract ω_o, we would subsequently be able to find all other frequencies and identify the corresponding defects.

3. METHODOLOGY

3.1 Dataset

The dataset used for training the model was obtained from [2]. Over a million data samples were used to train the model, each sample contains three features (i.e., frequency (Hz), PSD (g^2/Hz), and Load (%)) and one label (i.e., the Speed (rpm)). Table 2 shows some exemplary data samples. The data samples were obtained from experiments where the bearings were healthy, cone-defective, and cup-defective bearings. There were minimal data points for roller defects since the roller is the hardest component of the bearing assembly, and it hardly develops spalls. The objective is to train a ML model that can be used to predict the speed (dependent variable) from the input feature (independent variables), i.e., frequency, PSD, and load.

3.2 Data preprocessing

The most important element with regards to training ML models is the data, hence, data preprocessing is a very crucial step for programmers to acquire good results.

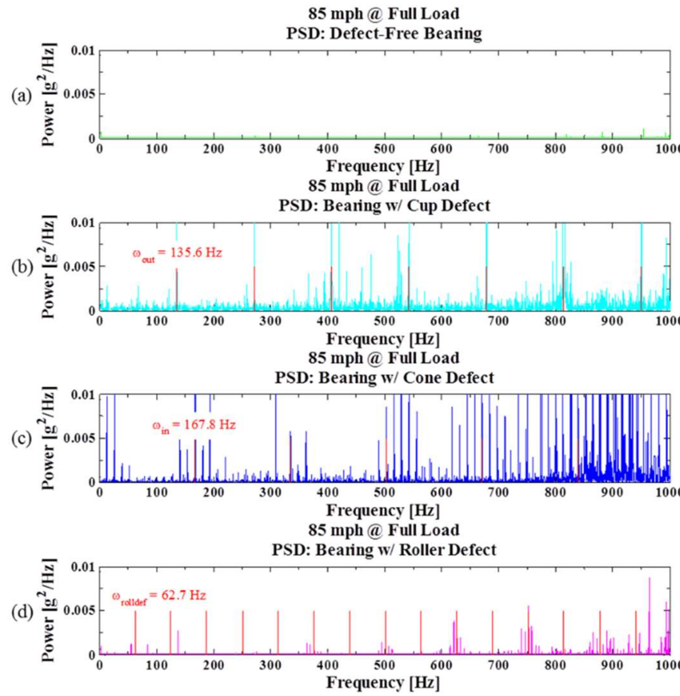

FIGURE 2: FREQUENCY SPECTRUM PLOTS (0-1000 Hz) OF (a) A DEFECT-FREE BEARING, (b) OUTER RING DEFECT, (c) INNER RING DEFECT, AND (d) A ROLLER DEFECT

The data was first filtered to take out rows with missing values. We then use the mean normalization to normalize the data to restrict all features in a similar scale to prevent certain features with larger magnitude from dominating the model. A random shuffle step was also adopted to prevent similar speeds from dominating the sets to give some variation. The data was then split into training and test sets in ratio of 70% and 30%, respectively.

TABLE 2: DATASET WITH INPUT FEATURES (FREQUENCY, LOAD, PSD) AND TARGET VALUE (SPEED)

Frequency [Hz]	PSD [g^2/Hz]	Load (%)	Speed (RPM)
0	2.72E-34	100	234
0.678168	3.06E-06	100	234
1.356337	4.60E-06	100	234
822.618273	0.000509075	100	514
823.296441	0.000165492	17	514
823.974609	0.000162635	17	514
824.652778	0.00036822	17	514

3.3 Model Training

The model used to train this data sample is a linear kernel method. To help understand KRR, let us consider a dataset $\{\boldsymbol{x}_i,\ y_i\}_{i=1}^{n}$ with $x_i \in \mathbb{R}^m$ represents the features with i-th data sample y_i denotes the output values. Let $\mathbf{y} = [y_1, \dots, y_n] \in \mathbb{R}^n$ and $\mathrm{X} = [\boldsymbol{x}_1; \dots; \boldsymbol{x}_n] \in \mathbb{R}^{n\times m}$. Assuming there is a linear relationship between y and X such that

$$\hat{\boldsymbol{y}} = X\boldsymbol{\beta}, \quad (8)$$

where $\hat{\boldsymbol{y}} = [\hat{y}_1, \dots, \hat{y}_n] \in \mathbb{R}^n$ represents the predicted output and $\boldsymbol{\beta} \in \mathbb{R}^m$ is the vector of coefficients. The traditional ridge regression solves the following optimization problem:

$$min_\beta \left|\left|\boldsymbol{y} - X\boldsymbol{\beta}\right|\right|_2^2 + \lambda\left|\left|\boldsymbol{\beta}\right|\right|_2^2. \quad (9)$$

Here λ is the coefficient that controls overfitting.

However, it is usually not practical to have linear assumptions in practice. To combat this problem, we seek to find a nonlinear model that best describes the relationship between each data pairs utilizing the kernel methods. To be specific, we are trying to find a nonlinear function that can be represented by the weighted kernel expansion over the data samples as follows:

$$f(\boldsymbol{x}) = \sum \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{k}_X(\boldsymbol{x}) \quad (10)$$

where $\boldsymbol{k}_X(\boldsymbol{x}) \in \mathbb{R}^n$ collects all $k(\boldsymbol{x}_i, \boldsymbol{x})$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the new parameter to be learned. In our paper, we use a linear kernel, which is a special case of a polynomial kernel with degree =1 and coefficient = 0 (homogeneous). If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are column vectors, their linear kernel is:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i{}^T\boldsymbol{x}_j\,. \quad (11)$$

Kernel ridge regression is thus a nonlinear case of traditional ridge regression that utilizes the functional mapping capabilities of kernel methods. The optimization function (9) thus becomes.

$$min_{\boldsymbol{\alpha}} \left|\left|\boldsymbol{y} - \boldsymbol{K\alpha}\right|\right|_2^2 + \lambda\left|\left|\boldsymbol{\alpha}^T\boldsymbol{K\alpha}\right|\right|_2^2, \quad (12)$$

with $\boldsymbol{K}$ collecting all $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for the data samples that we use to train the model.

4. RESULTS AND DISCUSSION

We utilize linear kernel method to train the model and Table 3 shows a comparison of the Root Mean Squared Error (RMSE*)* and Mean Absolute Error (MAE) for the train and test data. RMSE and MAE are both metrics used to evaluate the performance of regression models, but they measure the errors between predicted values and actual values differently.

RMSE is calculated by taking the square root of the average of squared differences between predicted and actual values. It gives higher weight to larger errors because of the squaring operation. RMSE is sensitive to outliers as it squares the errors, making it more influenced by large deviations between predicted and actual values. The equation below shows how the RMSE is calculated.

$$RMSE = \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\,, \quad (13)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

The MAE is calculated by taking the average of the absolute differences between predicted and actual values. It treats all errors equally and is less sensitive to outliers compared to RMSE.

$$MAE = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}|y_i - \hat{y}_i|\quad. \quad (14)$$

The model shows consistency (results are same for both train and test sets) in the performance due to the same values of RMSE and MAE for both train and test data. This indicates that the model is not overfitting or underfitting badly. The model is producing similar results from seen and unseen data. The RMSE = 102.4167 from TABLE 3 indicates that the average error of the predictions is approximately 102 units, considering the scale of the target variable. The MAE = 57.9806 from Table 3 suggests that, on average, the predicted values have an absolute difference of approximately 58 units from the actual values. In practical terms, the model's predictions deviate by approximately 58 units from the actual values in the test set.

TABLE 3: TRAIN AND TEST DATA LOSS COMPARISON

Metric	**Train**	**Test**
RMSE	99.7125	102.4167
MAE	56.0269	57.9806

In order to get a better understanding of the RMSE values we can scale them down using the formula:

$$RMSE_{scaled} = \frac{RMSE}{SPEED_{MAX} - SPEED_{MIN}} \quad (15)$$

where $RMSE = 102.4167$

$$SPEED_{MAX} = 618$$

$$SPEED_{MIN} = 234$$

This gives a $RMSE_{scaled} = 0.2667$, that shows that the predicted values show an error of 26.7% from the expected values. The relatively high RMSE and MAE is because of the sharp deviations in speed values in our train and test set as shown in Figure 3.

Initially, it is observed that the linear kernel method does well in predicting the pattern of the expected values but shows an error in prediction when there is a peak in values expected. This is because the linear kernel method is unable to capture the non-linear behavior present in the dataset. With speed values ranging from 234 to 618 rpm there are likely several of these deviations in our dataset resulting to a relatively high value for RMSE and MAE.

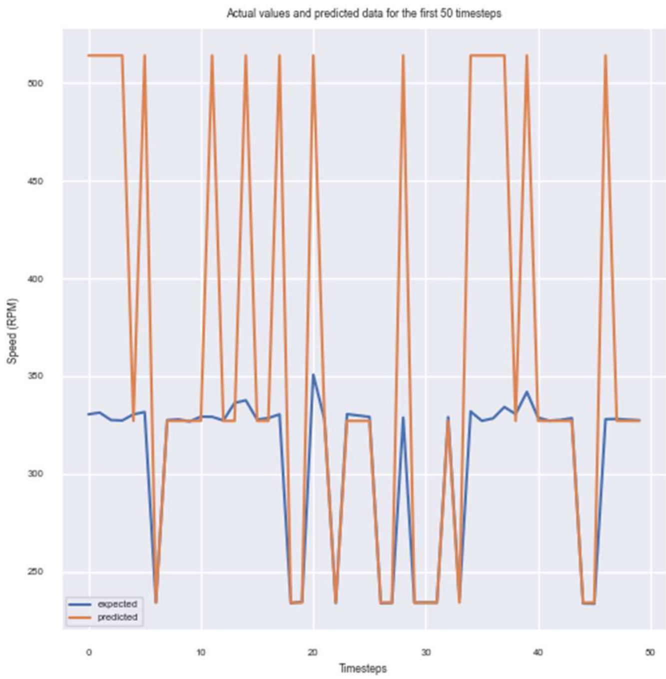

FIGURE 3: COMPARISON BETWEEN EXPECTED AND PREDICTED VALUES OF THE SPEED (RPM)

5. CONCLUSION

In summary, the model shows consistency between train and test performance. The range of speed values in rpm for trains (234 -618rpm) is relatively bigger than that in mph (25 – 66mph), hence training the model with speed values in mph with more features will result in a better prediction and a smaller RMSE and MAE values. Additionally, further exploration with different models such as Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) will be carried out to compare their performances to the Linear kernel method.

ACKNOWLEDGEMENTS

The authors want to acknowledge Hum Industrial Technology, Inc. for supporting this work. The authors also want to acknowledge the University Transportation Center for Railway Safety (UTCRS) at UTRGV for the financial support provided to perform this study through the USDOT UTC Program under Grant No. 69A3552348340.

REFERENCES

[1] S. Riaz, H. Elahi, K. Javaid, and T. Shahzad, "Vibration Feature Extraction and Analysis for Fault Diagnosis of Rotating Machinery-A Literature Survey," *Asia Pacific Journal of Multidisciplinary Research*, vol. 5, no. 1, pp. 103–110, 2017, Accessed: Nov. 30, 2023. [Online]. Available: www.apjmr.com

[2] J. Montalvo, C. Tarawneh, and A. A. Fuentes, "Vibration-Based Defect Detection for Freight Railcar Tapered-Roller Bearings," *2018 Joint Rail Conference*, Jun. 2018, doi: 10.1115/JRC2018-6210.

[3] Hofmann, Thomas, Bernhard Schölkopf, and Alexander J. Smola. "Kernel Methods In Machine Learning." (2008): 1171-1220.

[4] Steinwart, Ingo, and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

[5] Müller, Berndt, Joachim Reinhardt, and Michael T. Strickland. *Neural Networks: an Introduction*. Springer Science & Business Media, 2012.